# RL - Assignment 3

Srivatsava Kesanupalli

MT18054

Question - 3 Exercise 5.6

Equation analogous to $V(s) = \dfrac{\sum_{t\in\mathcal{J}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t\in\mathcal{J}(s)} \rho_{t:T(t)-1}}$ :

for action values $Q(s,a)$

we have $q_\pi(s,a) = E\left[\rho_{t:T-1} G_t \mid S_t = s, A_t = a\right]$

To estimate $q_\pi(s,a)$ we scale the returns by ratios and average the results.

$$Q(s,a) = \frac{\sum_{t\in\mathcal{J}(s,a)} \rho_{t:T(t)-1} G_t}{\sum_{t\in\mathcal{J}(s,a)} \rho_{t:T(t)-1}}$$

$$\rho_{t:T-1} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)} \rightarrow \text{importance sampling ratio}$$

Question - 5 Exercise 6.2

Why Temporal Difference methods are more efficient than Monte Carlo methods. We here consider the driving home example.

Now that we have moved to a new building and new parking lot, we still enter the highway at the same place. This is because, the ~~action~~ values for many intermediate states remain the same. Consider, the old ~~action~~ values of the states in between the workplace and home be the true values for current trajectory, TD moves close to true values than Monte Carlo does. This phenomena can be explained by the following equations

$$\text{TD} \quad V(S_t) \leftarrow \cancel{\alpha} V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) \mp V(S_t)]$$

$$\text{MC} \quad V(S_t) \longleftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

TD updates the values before the episode is completed owing to the $\gamma V(S_{t+1})$ term

For MC, this is not true as it waits for the episode to complete.

Question-8 Exercise 6.12

Q-Learning vs. Sarsa

Q-Learning

Initialize $Q(s,a); \forall\ s \in S^+; a \in A(s)$

for each episode:

Initialize S; for each step of episode {

Choose A from S using policy derived from Q

Observe R, S' from A

$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$

$S \leftarrow S'$ }

until S is terminal

Sarsa:

Initialize $Q(s,a); \forall\ s \in S^+; a \in A(S)$

for each episode:

Initialize S

for each step of episode:

Observe R, S' from A

Choose A' from S' using policy derived from Q

$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$

$S \leftarrow S'$; $A \leftarrow A'$

until S is terminal

Since Q-learning is an off-policy algorithm, so it chooses the best policy which satisfies the convergence conditions. Also, from the pseudocode it can be seen that the Q-function is updated first and then the action is chosen in the next iteration. Hence, the weight updates are differerd.

Sarsa on the other hand is an on-policy algorithm, it choses the next action and updates the weights based on the new states. The Q-function is updated after the action is obtained. The weight updates are different from that of Q-learn-ing and hence, both are not same although both the algorithm-s converge at one point after indefinite amount of time.

Question 6:

6.3, 6.4, and 6.5

6.3:

We have $V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$

$\alpha = 0.1$

$\gamma = 1$

initially we begin with value = 0.5 for every state

$\therefore V(A) = 0.5$ initially

On completing the first episode

V(A) is updated to the following on reaching the terminal state

$$V(A) \leftarrow 0.5 + (0.1)[\underbrace{0}_{\text{reward on reaching terminal state}} + 1(0) - 0.5]$$

$$V(A) \leftarrow 0.5 - 0.05$$
$$= 0.45$$

Which is what can be observed. The values for other states remain the same as the next state (be it left or right) is equal to itself and hence $V(S_{t+1})$ and $V(S_t)$ cancel out

## 6.4

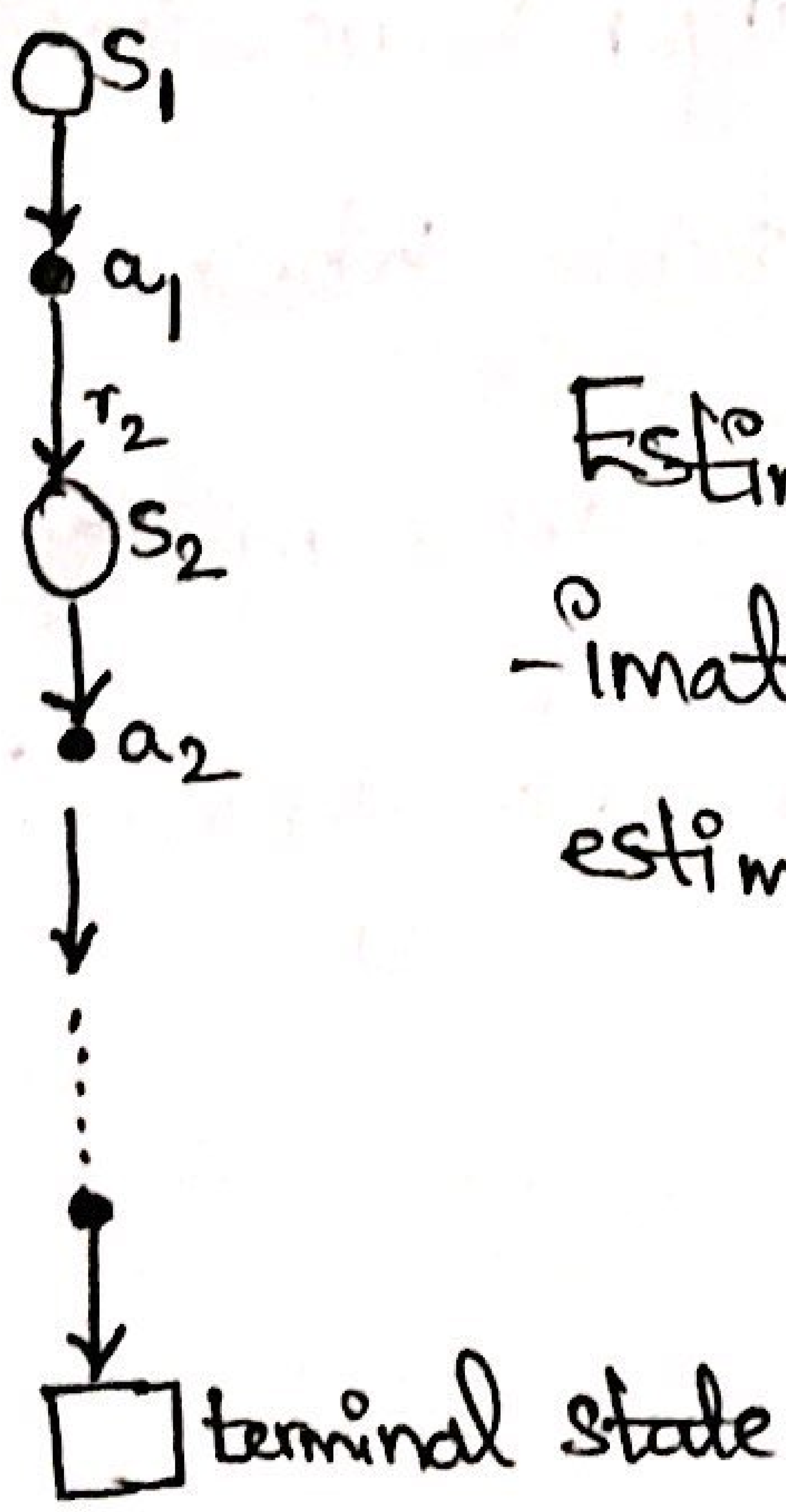
Smaller values of $\alpha$ are better than larger values in the long run as the learning is better. To decide which among the two algorithms viz., MC and TD, wider range of alpha values cannot give a conclusive evidence. In general, TD converges better than MC

## 6.5

With higher values of $\alpha$, the change in action value estimates is high and hence a major difference in exploration. Also, higher $\alpha$'s won't allow much learning. Initially, since the action values are high different actions are explored but this doesnot continue. Especially with TD which highly depends on the returns from new action value pairs, the root mean squared error is high with growing episodes. Lower $\alpha$'s explore better and the rms error although higher in the beginning, gradually reduces with growing episodes.

## Question-2

Backup diagram for MC estimation of $q_\pi$




Estimates for each state are independent and es
-imate for one state doesnot build upon the estimate of another state.

Question 1 : Rewrite the pseudo code ~~for~~ for MC Exploring starts

Importantly in MC ES we have the below block

for each episode ~~t = T-1, T-2, ..... 0~~

~~Generate~~ Choose $S_0 \in S$ and $A_0 \in A$ randomly

Generate an episode from $S_0, A_0$

$G \leftarrow 0$ [returns from the episode]

for each step of episode i.e $t = T-1, T-2, \ldots 0$

$G \leftarrow \gamma G + R_{t+1}$

Unless $S_t, A_t$ appears in $S_0, A_0, S_1, A_1 \ldots$ :

Append $G$ to Returns$(S_t, A_t)$

$Q(S_t, A_t) \leftarrow$ average (Returns$(S_t, A_t)$)

$\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$

This step involves calculating mean multiple times

According to section 2.4 incremental updates

$$Q(S_t, A_t) = \frac{1}{T} \sum_{i=t}^{T} G_i$$

$$= \frac{1}{T}\left[G_{T-1} + \frac{1}{(T-1)}(T-1)\sum_{i=t}^{T-1} G_{t-1}\right]$$

$$= \frac{1}{T}\left[G_{T-1} + (T-1)\, Q(S_{t-1}, A_{t-1})\right]$$

which is finally $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{T}\left(G_t - Q(S_t, A_t)\right)$

replace it in the averaging step and we have the redundant step removed.